WHAT IS CLAIMED IS:

1. A method for sharing a subinstruction of a given instruction among functional processing units of a plurality of clusters on a processor having a very long instruction word architecture, the given instruction including a set of control bits and at least one subinstruction, the processor comprising the plurality of clusters, each one cluster of the plurality of clusters comprising a plurality of functional processing units, the method comprising the steps of:

testing the set of control bits to identify a prescribed condition;

when the prescribed condition is identified, routing said subinstruction of the given instruction to multiple functional processing units as determined by the prescribed condition;

concurrently executing the subinstruction at said multiple functional processing units.

2. The method of claim 1, in which the step of routing comprises routing said subinstruction of the given instruction to a first functional processing unit of a first cluster of the plurality of clusters and to a first functional processing unit of a second cluster of the plurality of clusters; and in which the step of executing comprises concurrently executing the subinstruction at said first functional processing unit of the first cluster of the plurality of clusters and to the first functional processing unit of the second cluster of the plurality of clusters.

3. The method of claim 2, in which the given instruction comprises a first subinstruction and a second subinstruction, the step of testing comprising testing the set of control bits to identify a first prescribed condition, the step of routing comprising routing the first subinstruction, the method further comprising the steps of:

testing the set of control bits to identify a second prescribed condition;

when the second prescribed condition is identified, routing said second subinstruction of the given instruction to a second functional processing unit of the first cluster of the plurality of clusters and to a second functional processing unit of the second cluster of the plurality of clusters; and

concurrently executing the subinstruction at the first functional processing unit and the second functional processing unit; and

wherein the step of executing comprises concurrently executing the first subinstruction at the first functional processing unit of the first cluster, the first

subinstruction at the first functional processing unit of the second cluster, the second subinstruction at the second functional processing unit of the first cluster and the second subinstruction at the second functional processing unit of the second cluster.

4. A method for storing an instruction of a computer program to be executed on a processor having a very long instruction word architecture,

wherein each instruction comprises at least one subinstruction and up to a first prescribed number of subinstructions, the first prescribed number being at least two,

wherein the processor is organized into a plurality of clusters equaling a second prescribed number, each one cluster of the plurality of clusters comprising a common number of functional processing units, wherein the common number of functional processing units times the second prescribed number equals the first prescribed number,

wherein for a given instruction having the first prescribed number of subinstructions, each functional processing unit of the plurality of clusters is for executing a respective subinstruction of the given instruction, the method comprising the steps of:

identifying a pattern in which a subinstruction occurs more than once in the given instruction, said subinstruction being a redundant subinstruction;

determining whether the pattern is among a set of prescribed patterns;

when the pattern is among the set of prescribed patterns, setting a set of control bits for the instruction to indicate that said pattern is present.

5. The method of claim 3, further comprising compressing the given instruction when the pattern is among the set of prescribed patterns by deleting one occurrence of the redundant subinstruction in the given instruction to achieve a compressed instruction.

6. The method of claim 5, further comprising the steps of:

moving the compressed instruction into instruction cache;

testing the set of control bits of the compressed instruction to determine a condition is identified in which subinstruction sharing is to occur for the compressed instruction;

Cont B1

when subinstruction sharing is determined to occur, parsing the compressed instruction to route the redundant subinstruction to a plurality of functional processing units as determined by the identified condition;

concurrently executing the subinstruction at said plurality of functional processing units.

7. A method for storing an instruction of a computer program to be executed on a processor having a very long instruction word architecture,

wherein each instruction comprises at least one subinstruction and up to a first prescribed number of subinstructions, the first prescribed number being at least four,

wherein the processor is organized into a plurality of clusters equaling a second prescribed number, each one cluster of the plurality of clusters comprising a common number of functional processing units, wherein the common number of functional processing units times the second prescribed number equals the first prescribed number,

wherein for a given instruction having the first prescribed number of subinstructions, each functional processing unit of the plurality of clusters is for executing a respective subinstruction of the given instruction, the method comprising the steps of:

for the given instruction, comparing a first subinstruction which is to be processed by a first functional unit of a first cluster of the plurality of clusters with a second subinstruction which is to be processed by a first functional unit of a second cluster of the plurality of clusters;

for a case in which the first subinstruction is the same as the second subinstruction setting a first control bit of a set of control bits associated with the given instruction to a first logic state which indicates that the second subinstruction equals the first subinstruction;

for the given instruction, comparing a third subinstruction which is to be processed by a second functional unit of the first cluster of the plurality of clusters with a fourth subinstruction which is to be processed by a second functional unit of the second cluster of the plurality of clusters; and

for a case in which the third subinstruction is the same as the fourth subinstruction setting a second control bit of the set of control bits associated with the given instruction to a second logic state which indicates that the fourth subinstruction equals the third subinstruction; and

storing the given instruction with the first control bit and the second control bit.

8. The method of claim 7, in which the step of storing comprises storing the given instruction in an uncompressed format, and further comprising the steps of compressing the given instruction into a compressed format, and storing the given instruction in cache in the compressed format, the step of compressing comprising the steps of:

testing the first control bit associated with the given instruction;

for a case in which the first control bit equals the first logic state compressing the given instruction to reduced size in which one copy of the equal first subinstruction and second subinstruction is omitted to avoid redundant storage of the first subinstruction and the second subinstruction;

testing the second control bit associated with the given instruction; and

for a case in which the second control bit equals the second logic state compressing the given instruction to reduced size in which one copy of the equal third subinstruction and fourth subinstruction is omitted to avoid redundant storage of the third subinstruction and the fourth subinstruction.

9. The method of claim 7, in which the step of storing comprises storing the given instruction in a compressed format, and further comprising prior to the step of storing, the step of compressing the given instruction into the compressed format, the step of compressing, comprising the steps of:

when the first control bit equals the first logic state compressing the given instruction to reduced size in which one copy of the equal first subinstruction and second subinstruction is omitted to avoid redundant storage of the first subinstruction and the second subinstruction;

when the second control bit equals the second logic state compressing the given instruction to reduced size in which one copy of the equal third subinstruction and fourth subinstruction is omitted to avoid redundant storage of the third subinstruction and the fourth subinstruction.

10. The method of claim 9, further comprising the step of storing the given instruction in cache in the compressed format.

11. The method of claim 7, further comprising the steps of:

storing the given instruction in cache in a compressed format with the first control bit and the second control bit, the compressed format combining the storage of the first subinstruction with the storage of the second subinstruction into a first combined storage when the first control bit is set to the first logic state, the compressed format combining the storage of the third subinstruction with the storage of the fourth subinstruction into a second combined storage when the second control bit is set to the second logic state;

testing the first control bit;

when the first control bit is set to the first logic state, routing a content of the first combined storage to the first functional processing unit of the first cluster and the first functional processing unit of the second cluster for concurrent execution by the first functional processing unit of the first cluster and the first functional processing unit of the second cluster;

testing the second control bit; and

when the second control bit is set to the second logic state, routing a content of the second combined storage to the second functional processing unit of the first cluster and the second functional processing unit of the second cluster for concurrent execution by the second functional processing unit of the first cluster and the second functional processing unit of the second cluster.

12. A method for compressing into a compressed format, an instruction of a computer program to be executed on a processor having a very long instruction word architecture,

wherein each instruction comprises at least one subinstruction and up to a first prescribed number of subinstructions, the first prescribed number being at least four,

wherein the processor is organized into a plurality of clusters equaling a second prescribed number, each one cluster of the plurality of clusters comprising a common number of functional processing units, wherein the common number of functional processing units times the second prescribed number equals the first prescribed number,

wherein for a given instruction having the first prescribed number of subinstructions, each functional processing unit of the plurality of clusters is for executing a respective subinstruction of the given instruction, the method comprising the steps of:

for the given instruction, comparing a first subinstruction which is to be processed by a first functional unit of a first cluster of the plurality of clusters with a second subinstruction which is to be processed by a first functional unit of a second cluster of the plurality of clusters;

for a case in which the first subinstruction is the same as the second subinstruction compressing the given instruction to be stored with the first subinstruction and without the second subinstruction, and setting a first control bit associated with the given instruction to a logic state which indicates that the second subinstruction equals the first subinstruction;

for the given instruction, comparing a third subinstruction which is to be processed by a second functional unit of the first cluster of the plurality of clusters with a fourth subinstruction which is to be processed by a second functional unit of the second cluster of the plurality of clusters; and

for a case in which the third subinstruction is the same as the fourth subinstruction compressing the given instruction to be stored with the third subinstruction and without the fourth subinstruction, and setting a second control bit associated with the given instruction to a logic state which indicates that the fourth subinstruction equals the third subinstruction.

13. A computer system comprising:

a processor having a very large word instruction architecture and including a plurality of clusters of functional processing units, each one cluster of the plurality of clusters comprising a common number of functional processing units, the processor comprising a first prescribed number of clusters, said very large word instruction architecture allowing an instruction to have up to a second prescribed number of subinstructions, where the second prescribed number equals the first prescribed number times the common number, each instruction to be executed by the processor comprising from one subinstruction up to the second prescribed number of subinstructions, along with a set of control bits; and

an instruction cache memory which stores a first instruction in a compressed format determined by a condition of the set of control bits, the compressed format including a shared subinstruction stored in a given field of the first instruction which is to be shared by a plurality of the functional processing units, said plurality of functional processing units being determined by said condition of the set of control bits.

14. The system of claim 13, in which said shared subinstruction is for a first functional processing unit of a first cluster and a first functional processing unit of a second cluster when the set of control bits identifies a first prescribed condition.

15. The system of claim 14, in which the shared subinstruction is a first shared subinstruction, and in which the compressed format further includes a second shared subinstruction for a second functional processing unit of the first cluster and a second functional processing unit of the second cluster when the set of control bits either concurrently identifies a second prescribed condition.

16. The system of claim 14, further comprising:

means for testing the set of control bits for a given instruction; and

means for routing said first common subinstruction to the first functional processing unit of the first cluster and to the first functional processing unit of the second cluster of the plurality of clusters when said testing means identifies the first prescribed condition.

17. The system of claim 16, in which the first common subinstruction is concurrently executed at the first functional processing unit of the first cluster and the first functional processing unit of the second cluster.

18. The system of claim 14, in which the first instruction in an uncompressed format includes the second prescribed number of subinstructions, the first instruction comprising a first subinstruction for being executed by a first functional processing unit of a first cluster and a second subinstruction for being executed by a first functional processing unit of a second cluster, the system further comprising means for compiling the first instruction, the compiling means comprising:

means for comparing the first subinstruction and the second subinstruction;

means for setting a state of the set of control bits to identify a first prescribed condition when the first subinstruction is equal to the second subinstruction.

Cont B1

19. The system of claim 14, in which a first instruction in uncompressed format includes the second prescribed number of subinstructions, the first instruction comprising a first subinstruction for being executed by a first functional processing unit of a first cluster and a second subinstruction for being executed by a first functional processing unit of a second cluster, the system further comprising means for compressing the first instruction into the compressed format, the compressing means comprising:

means for testing the set of control bits associated with the first instruction;

means for reducing the size of the first instruction by omitting the second subinstruction when the set of control bits identifies that the first subinstruction equals the second subinstruction.

20. The system of claim 14, in which a first instruction in uncompressed format includes the second prescribed number of subinstructions, the first instruction comprising a first subinstruction for being executed by a first functional processing unit of a first cluster and a second subinstruction for being executed by a first functional processing unit of a second cluster, the system further comprising means for caching the first instruction, the caching means comprising:

means for testing the set of control bits associated with the first instruction;

means for reducing the size of the first instruction to achieve a compressed format by omitting the second subinstruction when the set of control bits identifies that the first subinstruction equals the second subinstruction; and

means for loading the first instruction into the instruction cache in the compressed format.

21. The system of claim 14, in which a first instruction in uncompressed format includes the second prescribed number of subinstructions, the first instruction comprising a first subinstruction for being executed by a first functional processing unit of a first cluster and a second subinstruction for being executed by a first functional processing unit of a second cluster, the system further comprising means for caching the first instruction, the caching means comprising:

means for comparing the first subinstruction and the second subinstruction;

means for setting a state of the set of control bits associated with the first instruction to identify a first prescribed condition when the first subinstruction is equal to the second subinstruction.

means for reducing the size of the first instruction to achieve a compressed format by omitting the second subinstruction when the set of control bits identifies that the first subinstruction equals the second subinstruction; and

means for loading the first instruction into the instruction cache in the compressed format.